Cardiovascular Risk Prediction

Akash K,
Data science Trainee,
AlmaBetter, Bangalore.

Abstract:

The independent variables such as age, education, is_smoking etc… are the determinants of the dependent variable 'TenYearCHD'. I was provided with already classified labels in our data set.

Our experiment can help understand what could be the reason for the classification of such labels by feature selection, data analysis and prediction with machine learning algorithms taking into account previous trends to determine the correct classification.

Keywords:machine learning,TenYearCHD,support vector machines,classified labels

1.Problem Statement

The dataset is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. The classification goal is to predict whether the patient has a 10-year risk of future coronary heart disease (CHD). The dataset provides the patients' information. It includes over 4,000 records and 15 attributes. Variables

Each attribute is a potential risk factor. There are both demographic, behavioral, and medical risk factors.

2. Introduction

The TenCHD is predicted using the following Independent Variables:

• **Sex:** male or female("M" or "F")

• **Age:** Age of the patient;(Continuous - Although the recorded ages have been truncated to whole numbers, the concept of age is continuous)

Behavioral:

• **is_smoking:** whether or not the patient is a current smoker ("YES" or "NO")

• **Cigs Per Day:** the number of cigarettes that the person smoked on average in one day.(can be considered continuous as one can have any number of cigarettes, even half a cigarette.)

Medical(history):

• **BP Meds:** whether or not the patient was on blood pressure medication (Nominal)

• **Prevalent Stroke:** whether or not the patient had previously had a stroke (Nominal)

• **Prevalent Hyp:** whether or not the patient was hypertensive (Nominal)

• **Diabetes:** whether or not the patient had diabetes (Nominal)

Medical(current):

• **Tot Chol:** total cholesterol level (Continuous)

• **Sys BP:** systolic blood pressure (Continuous)

• **Dia BP:** diastolic blood pressure (Continuous)

• **BMI:** Body Mass Index (Continuous)

• **Heart Rate:** heart rate (Continuous - In medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of a large number of possible values.)

• **Glucose**: glucose level (Continuous)

Predict variable (desired target):

• **TenCHD:** 10-year risk of coronary heart disease CHD(binary: "1", means "Yes", "0" means "No") - **DV**

Our goal here is to build a predictive model, which could help the hospitals in predicting Chronic Heart Disease proactively.

3. Steps involved:

- **Exploratory Data Analysis**
  After loading the dataset I performed this method by comparing our target variable that is TenCHD with other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

- **Null values Treatment**
  Our dataset contains a large number of null values which might tend to disturb our accuracy hence I dropped them at the beginning of our project inorder to get a better result.

- **Encoding of categorical columns**
  I used One Hot Encoding to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

- **Feature Selection**
  In these steps I used algorithms like ExtraTree classifier to check the results of each feature i.e which feature is more important compared to our model and which is of less importance.

- **Standardization of features**
  Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.
  The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

- **Fitting different models**
  For modelling I tried various classification algorithms like:

1. **Logistic Regression**
2. **SVM Classifier**
3. **Decision Trees**
4. **Random Forest Classifier**
5. **Gradient Boosting Machine**
6. **XGBoost classifier**

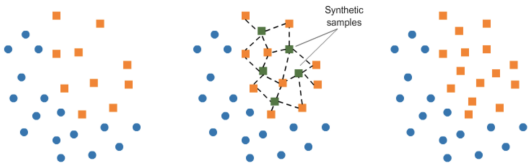
- **<u>Tuning the hyperparameters for better accuracy</u>**
  Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in case of tree based models like Random Forest Classifier and XGBoost classifier.

- **<u>SHAP Values for features</u>**
  I have applied SHAP value plots on the Random Forest model to determine the features that were most important while model building and the features that didn't put much weight on the performance of our model.

4. Application of SMOTE to the Dataset:

This technique generates synthetic data for the minority class.

SMOTE (Synthetic Minority Oversampling Technique) works by randomly picking a point from the minority class and computing the k-nearest neighbors for this point. The synthetic points are added between the chosen point and its neighbors.



SMOTE algorithm works in 4 simple steps:

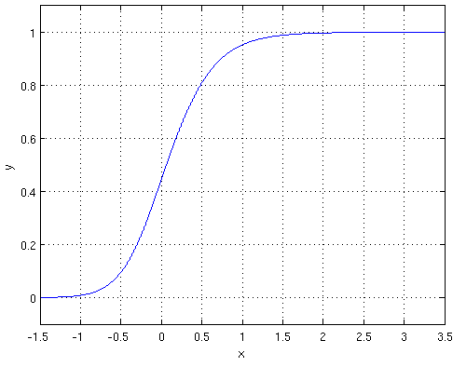
- Choose a minority class as the input vector
- Find its k nearest neighbors (k_neighbors is specified as an argument in the SMOTE() function)
- Choose one of these neighbors and place a synthetic point anywhere on the line joining the point under consideration and its chosen neighbor
- Repeat the steps until data is balanced

5.1. Algorithms:

1. Logistic Regression:

Logistic Regression is actually a classification algorithm that was given the name regression due to the fact that the mathematical formulation is very similar to linear regression.
The function used in Logistic Regression is sigmoid function or the logistic function given by:

f(x)= 1/1+e ^(-x)



The optimization algorithm used is: Maximum Log Likelihood. I mostly take log likelihood in Logistic:

$$\ln L(\mathbf{y},\beta)=\ln\prod_{i=1}^{n} f_i(y_i)=\sum_{i=1}^{n}\left[y_i \ln\left(\frac{\pi_i}{1-\pi_i}\right)\right]+\sum_{i=1}^{n}\ln(1-\pi_i)$$

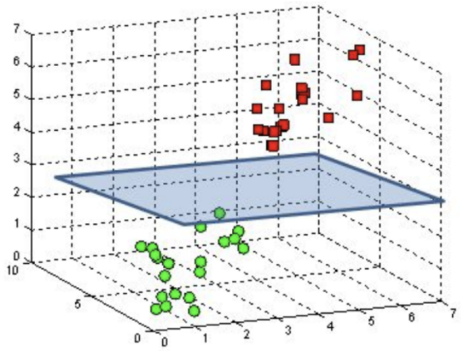
.

2. Support Vector Machine Classifier:

SVM is used mostly when the data cannot be linearly separated by logistic regression and the data has noise. This can be done by separating the data with a hyperplane at a higher order dimension.
In SVM I use the optimization algorithm as:

$$\min_{\xi,w,b} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$
$$\text{s.t} \quad y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 -$$
$$\xi_i \geq 0;\ i = 1,\ldots,m.$$

where C is a cost parameter and ξ_i's are slack variables.

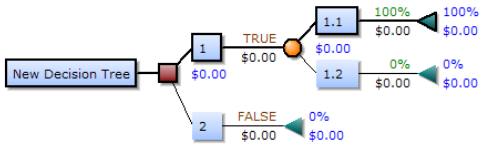


I use hinge loss to deal with the noise when the data isn't linearly separable.
Kernel functions can be used to map data to higher dimensions when there is inherent non linearity.

3. Decision Trees:

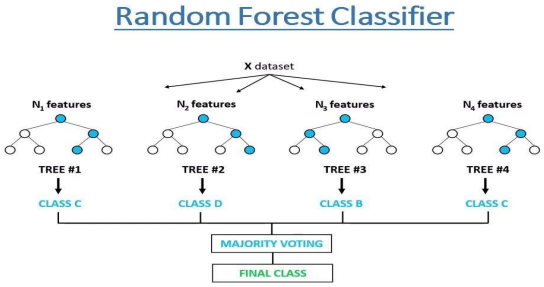
A **decision tree** is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.



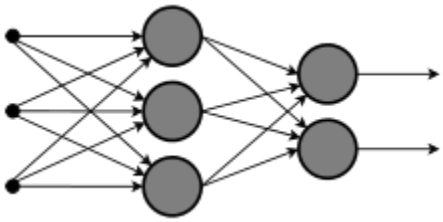
4. Random Forest Classifier:

Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the training set, collects the labels from these subsets and then averages the final prediction depending on the most number of times a label has been predicted out of all.



5. Gradient Boosting Machine:

Gradient boosting is a machine learning technique for regression, classification and other tasks, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient boosted trees, which usually outperforms random forest. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

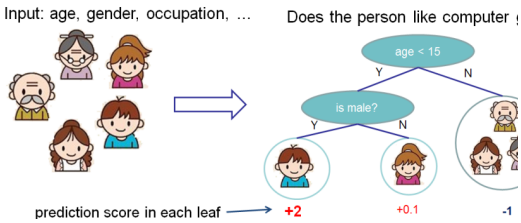


6. XGBoost-

To understand XGBoost I have to know gradient boosting beforehand.

- **Gradient Boosting-**

  Gradient boosted trees consider the special case where the simple model is a decision tree



  In this case, there are going to be 2 kinds of parameters P: the weights at each leaf, w, and the number of leaves T in each tree (so that in the above example, T=3 and w=[2, 0.1, -1]).

  When building a decision tree, a challenge is to decide how to split a current leaf. For instance, in the above image, how could I add another layer to the (age > 15) leaf? A 'greedy' way to do this is to consider every possible split on the remaining features (gender and occupation), and calculate the new loss for each split; you could then pick the tree which most reduces your loss.

XGBoost is one of the fastest implementations of gradient boosting. trees. It does this by tackling one of the major inefficiencies of gradient boosted trees: considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XGBoost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits.

5.2. Model performance:

Model can be evaluated by various metrics such as:

1. **Confusion Matrix-**

The confusion matrix is a table that summarizes how successful the classification modelis at predicting examples belonging to various classes. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.

2. **Precision/Recall-**

Precision is the ratio of correct positive predictions to the overall number of positive predictions : TP/TP+FP

Recall is the ratio of correct positive predictions to the overall number of positive examples in the set: TP/FN+TP

3. **Accuracy-**

Accuracy is given by the number of correctly classified examples divided by the total number
of classified examples. In terms of the confusion matrix, it is given by: TP+TN/TP+TN+FP+FN

4. **Area under ROC Curve(AUC)-**

ROC curves use a combination of the true positive rate (the proportion of positive examples predicted correctly, defined exactly as recall) and false positive rate (the proportion of negative examples predicted incorrectly) to build up a summary picture of the classification performance.

7.3. Hyper parameter tuning:

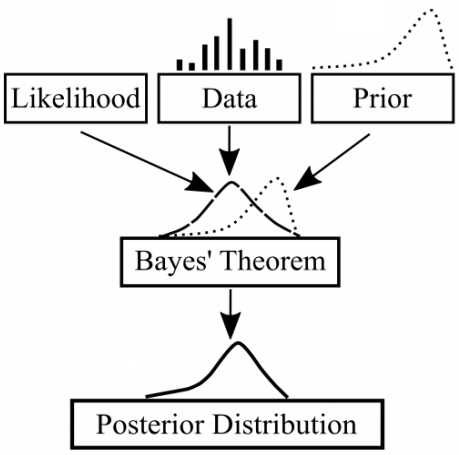
Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

I used Grid Search CV, Randomized Search CV and Bayesian Optimization for hyperparameter tuning. This also results in cross validation and in our case I divided the dataset into different folds. The best performance improvement among the three was by Bayesian Optimization.

1. **Grid Search CV-**Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model's performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

2. **Randomized Search CV-** In Random Search, the hyperparameters are chosen at random within a range of values that it can assume. The advantage of this method is that there is a greater chance of finding regions of the cost minimization space with more suitable hyperparameters, since the choice for each iteration is random. The disadvantage of this method is that the combination of hyperparameters is beyond the scientist's control.

3. **Bayesian Optimization-** Bayesian Hyperparameter optimization is a very efficient and interesting way to find good hyperparameters. In this approach, in naive interpretation way is to use a support model to find the best hyperparameters.A hyperparameter optimization process based on a probabilistic model, often Gaussian Process, will be used to find data from data observed in the later distribution of the performance of the given models or set of tested hyperparameters.



As it is a Bayesian process at each iteration, the distribution of the model's performance in relation to the hyperparameters used is evaluated and a new probability distribution is generated. With this distribution it is possible to make a more appropriate choice of the set of values that I will use so that our algorithm learns in the best possible way.

6. Conclusion:

That's it! I reached the end of our exercise. Starting with loading the data so far I have done EDA , null values treatment, encoding of categorical columns, feature selection and then model building.

In all of these models our accuracy revolves in the range of 85 to 90%.

And there is no such improvement in accuracy score even after hyperparameter tuning.

So the accuracy of our best model is 90% which can be said to be good for this large dataset. This performance could be due to various reasons like: no proper pattern of data, too much data, not enough relevant features.